

REMARKS

Applicants acknowledge receipt of the Examiner's Office Action dated January 6, 2009.

Claims **1-39** are pending in the application.

Claims **1-39** have been rejected.

Claims **1, 10, 25** and **34-38** have been amended.

Rejection of Claims under 35 U.S.C. § 112

Claims 34-39 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter that Applicant regards as the invention. Applicants have amended claims 34-38 to address the rejection and now submit that the claims are in a condition for allowance.

Rejection of Claims under 35 U.S.C. § 103

Claims 1-39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Marik, U.S. Patent No. 5,903,718 (Marik), in view of Akgul et al., U.S. Publication Application No. 2003/0074650 (Akgul), and in view of Fritz et al., U.S. Patent No. 7,296,187 (Fritz). Applicants respectfully traverse this rejection.

Independent claims 1, 10, 25, and 34 each contain features of substantially the following form:

“...
causing a debugger agent to select a debugger program suitable for a device under test,
wherein
the debugger agent is configured to select the debugger program from a plurality
of debugger programs, and
the device under test is configured to execute a program under test;
causing the debugger agent to load the debugger program into the device under test;
sending a plurality of test commands to the device under test according to a test script;
and
activating the debugger program when a watched event occurs during execution of the
program under test.”

Applicants respectfully submit that Marik, Akgul, nor Fritz, alone or in combination, teach or contemplate all of the features of claims 1, 10, 25, and 34 because Marik, Akgul, nor Fritz considers the process of selecting a debugger program or subsequently loading the debugger program onto the device under test.

As an initial matter, Marik's system does not allow a debugger program to be loaded, and with no loading possible, there can be no selection of a debugger program to load. Additionally, Fritz is cited only for introducing the concept of a script; however, the addition of the concept of a script does not cure the deficiencies of Marik, because adding scripts to Marik does not add the ability to select and load a debugger program onto the device under test.

Similarly, Akgul also fails to cure the deficiencies of Marik with respect to, among other limitations, the claimed process of selecting a debugger program, as well as the claimed subsequent loading of the debugger program onto the device under test. As an initial matter, Akgul fails to show, teach or suggest anything comparable to the claimed debugger agent and debugger program, at least because Akgul clearly fails to show, teach or suggest any sort of distinction between a debugger agent and a debugger program, in the manner of claim 1.

Instead, Akgul refers to a “Debugger OS” (Akgul, Figure 1) and a collection of debugger modules (Akgul, Figure 5), which are, in fact, the same construct. That these constructs are “one and the same” is evidenced in the Specification, which states that “[t]he Debugger OS 116 (such as one or more debugger modules of a debugger system) . . . reside in the memory installed on the target system.” Akgul, para. 33. That is, the Debugger OS is composed of the debugger modules. Thus, Akgul fails to show, teach or suggest a debugger agent that can select and load a debugger program, separate therefrom, into the device under test, as recited in claim 1.

Additionally, the system presented in Akgul operates in a manner that is fundamentally different than that of the claimed invention. Specifically, Akgul discloses a system in which a debugger module is selected and loaded onto a target system after a hardware exception occurs. Akgul, para. 42, 44. Akgul states, “When a hardware exception occurs, the system selects the base debugger modules that will be helpful to understand the base cause of the hardware exception and, if this option is enabled, asks the user’s permission to load them into the system dynamically.” Akgul para. 42 (Emphasis supplied). This is in direct contrast to the language of claim 1, in which the debugger agent loads the debugger program into the device under test, a plurality of test commands are sent to the device under test according to a test script, and then the debugger program is activated when a watched event occurs during execution of the program under test. Thus, in marked contrast to the claimed invention (e.g., claim 1, which teaches a system in which the debugger program is loaded into the device under test prior to a watched event occurs), a system according to Akgul waits until a hardware exception occurs before a base debugger module is loaded,.

Akgul goes on to make this distinction even more clear. Akgul states, “During error-free operation at runtime, the debugger modules are not typically loaded into the system . . . As soon as an error condition occurs (e.g., hardware rises an exception or a software assertion fires), the

debugger modules are loaded from the external storage unit.” Akgul, para. 44. This is unlike the debugger agent selecting a debugger program for loading into the device under test and subsequently being activated when a watched event occurs during execution of the program under test as taught in claim 1. Not only is the loading of debugger modules in Akgul predicated solely on an error condition, but the forms of error conditions included in the above quote are specifically tied to the hardware and software, respectively, of the target system being tested, and are not based upon a debugger agent as taught by claim 1. Of note, Akgul describes a hardware exceptions “are caught by the support of the underlying hardware.” Akgul, para. 41. Similarly, the software assertions used in the above Akgul quote are described as “assertions are good examples for the instrumentation code that can be added into the code being debugged.” Akgul, para. 43 (emphasis added). Thus, Akgul purports a system in which the debugger module may be selected based upon assertions that are added into the code being debugged. In contrast, claim 1 recites a debugger agent that selects a debugger program suitable for the device under test, after which a plurality of test commands are sent to the device under test according to a test script. The debugger program is then activated when a watched event occurs during execution of the program under test.

Further, Applicants respectfully submit that Marik and Fritz cannot be properly combined because such a combination would change the principle of operation of each system. As an initial matter, Applicants begin with the MPEP which states that “the claimed combination cannot change the principle of operation of the primary reference or render the reference inoperable for its intended purpose. See MPEP § 2143.01.” MPEP § 2145, section III.

Marik’s system design is presented in Figure 1, which depicts a PC Host directly connected to a target system and communication occurs over a serial connection and interrupt lines. On the other hand, Fritz explains that his system introduces a hardware debug device for

the specific purpose of reducing the traffic between a host and the device under test. *See* Fritz, Abstract. Fritz's system is specifically designed to avoid the problems extant in a communication environment such as that propounded by Marik. Thus, one of ordinary skill in the art would be discouraged from adding Marik to Fritz because such an addition would nullify any advantages afforded by Fritz. Marik's use of interrupt lines directly into the device under test would be thwarted by the introduction of Fritz's hardware debug device between the PC Host and the device under test. Because each system depends on a feature that would be inoperable in the environment provided by the other system, these two systems cannot be combined without nullifying critical features of each system. This nullification of critical features would change the principle of operation of each system and therefore these two references are not properly combinable.

CONCLUSION

Applicants submit that all claims are now in condition for allowance, and an early notice to that effect is earnestly solicited. Nonetheless, should any issues remain that might be subject to resolution through a telephonic interview, the Examiner is requested to telephone the undersigned.

If any extensions of time under 37 C.F.R. § 1.136(a) are required in order for this submission to be considered timely, Applicant hereby petitions for such extensions. Applicant also hereby authorizes that any fees due for such extensions or any other fee associated with this submission, as specified in 37 C.F.R. § 1.16 or § 1.17, be charged to deposit account 502306.

Respectfully submitted,

/ Samuel G. Campbell, III /

Samuel G. Campbell, III
Attorney for Applicants
Reg. No. 42,381
Telephone: (512) 439-5084
Facsimile: (512) 439-5099